v.5.0

ĆWICZENIE V

PROCESOR SYGNAŁOWY - WPROWADZENIE

(00)

Ćwiczenie jest krótkim wprowadzeniem do programowania procesora sygnałowego. Wykorzystuje się zintegrowane środowisko programistyczne: „Code Composer Studio" i „DSP Starter Kit" - układ wyposażony w procesor zmiennoprzecinkowy TMS320C6711 wraz z obsługą wejść i wyjść analogowych. Alternatywę środowiska programowania układu DSP stanowi program Simulink.

Wykonując ćwiczenie należy zwrócić uwagę na oznaczenia ikonami poszczególnych akapitów tej instrukcji. Mają one na celu usprawnienie wykonywania ćwiczeń.

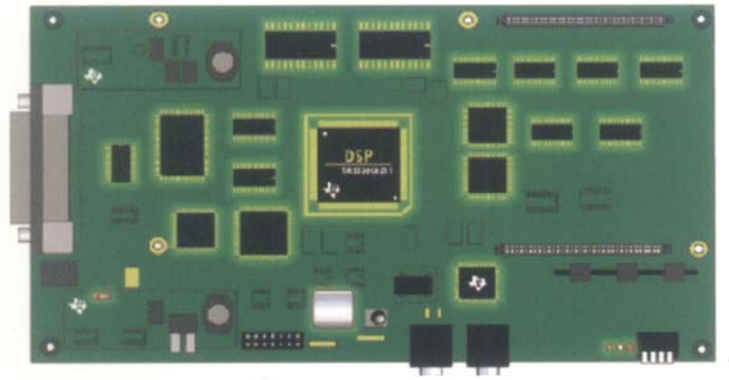
Oznaczenia

- Ⓟ przykłady
- ◈ opis funkcji MATLABA
- ↗ zadania do wykonania.
- **(15)** oznacza, iż aktualne zadanie powinno być wykonywane w czasie nie późniejszym niż 15 minuta ćwiczeń.

Czas wykonania ćwiczenia wynosi 180 minut.

WPROWADZENIE

Układ DSK

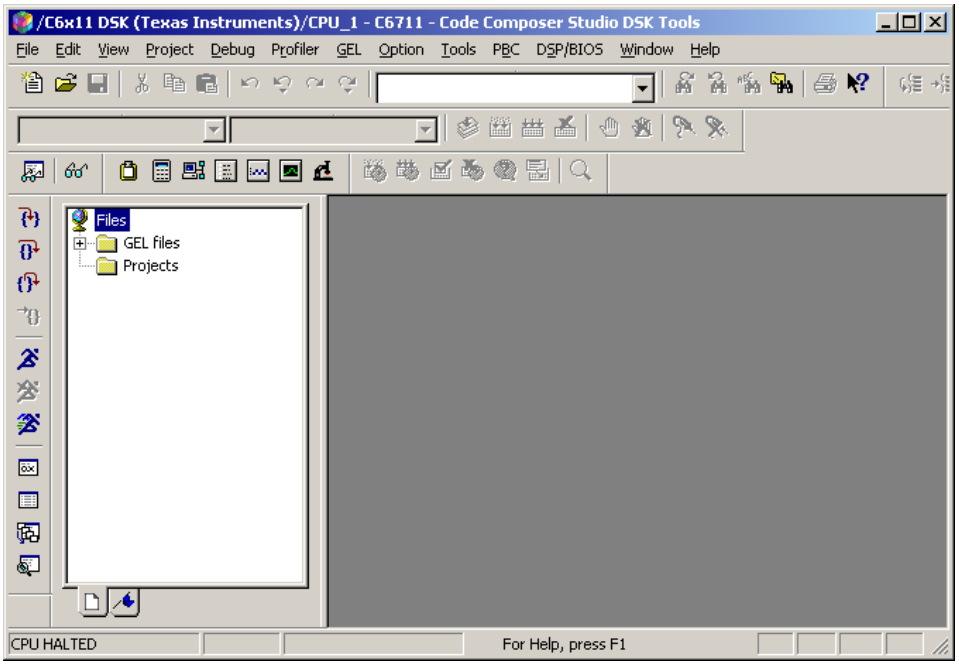


Rys.1. Układ DSK

Układ DSK zawiera zmiennoprzecinkowy procesor sygnałowy serii C6711 oraz 16-bitowy przetwornik AD535 dla wejścia i wyjścia. AD535 wykorzystuje do konwersji A/C i C/A technologię sigma-delta. Częstotliwość próbkowania jest stała i wynosi 8 kHz. Układ DSK wyposażono w 16MB pamięć RAM oraz 128kB pamięć flash ROM.

Procesor sygnałowy C6711 bazuje na architekturze VLIW, (very-long-instruction-word), która jest odpowiednia w przypadku bardzo intensywnych algorytmów numerycznych. W każdym cyklu zegarowym (co 6,6ns) jest wykonywanych osiem 32-bitowych instrukcji.

CODE COMPOSER STUDIO



Rys. 2. Program CCS

Code Composer Studio (CCS) jest to zintegrowane środowisko programistyczne (IDE), zawierające takie narzędzia jak kompilator języka C, assembler oraz linker. CCS pozwala na edycję programów oraz ich debug'owanie w czasie rzeczywistym.

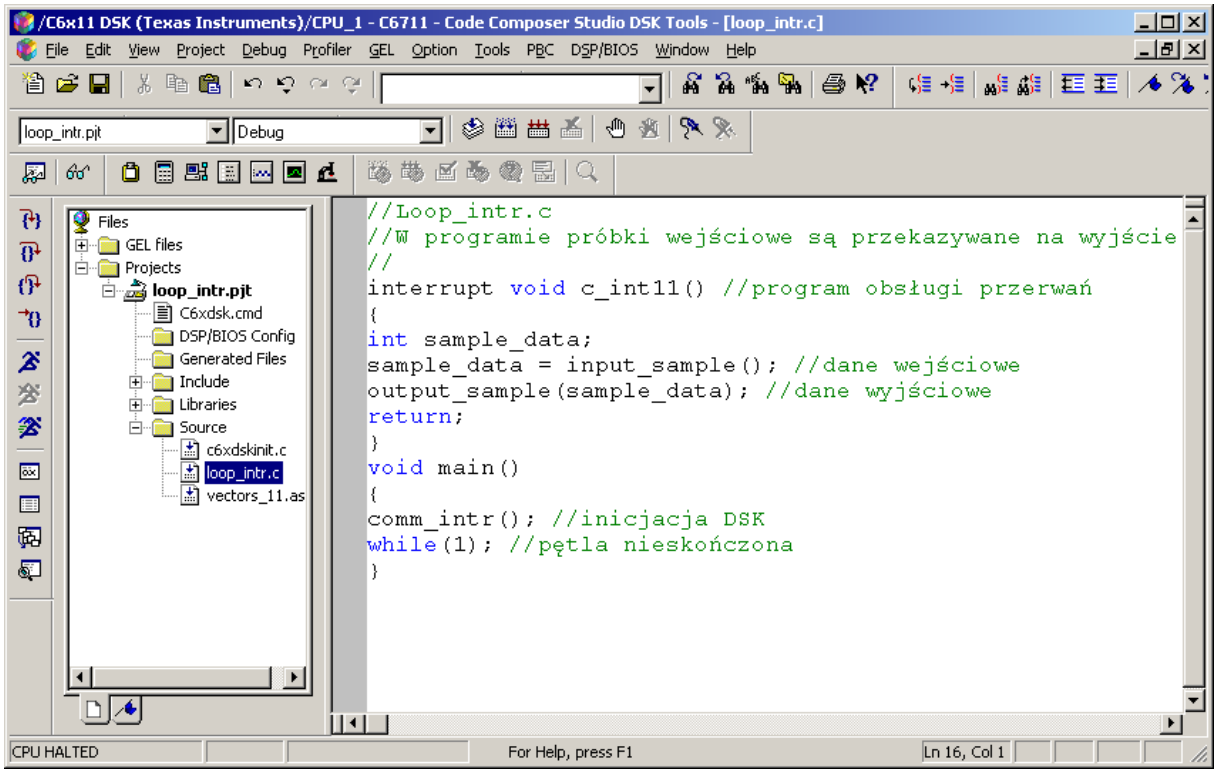
1. PROJEKT PĘTLA

Projekt ten ma dwa cele: wprowadzenie do programowania układu procesora sygnałowego przy wykorzystaniu środowiska CCS oraz sprawdzenie niektórych charakterystycznych cech układu DSK.

Aby rozpocząć pracę należy sprawdzić czy wszystkie połączenia urządzeń na stanowisku laboratoryjnym są wykonane prawidłowe:

- komputer - złącze równoległe - DSK
- generator - chinch - DSK
- DSK - chinch - oscyloskop

Przed uruchomieniem CCS należy usunąć wszystkie podkatalogi z **c:\ti\myprojects**.
Pobrać archiwum zawierające projekty do ćwiczenia (**cw5proj.zip lub cw5proj.exe**) i rozpakować je w katalogu **c:\ti\myprojects**
Uruchomić CCS oraz **wczytać** projekt 1 z katalogu **c:\ti\myprojects\projekt1** wybierając opcje: **Project –Open-**(otwórz plik: **loop_intr.pjt**)



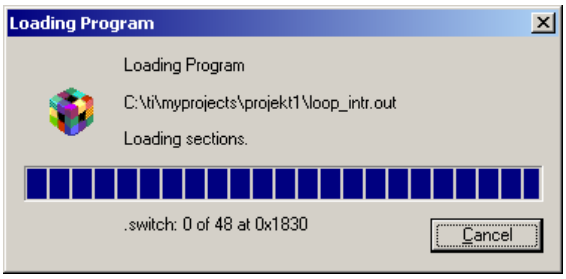
Po lewej stronie CCS znajduje się zapis całej struktury projektu w postaci katalogów, w których zawarte są wszystkie potrzebne pliki (biblioteki i pliki konfiguracyjne) do skompilowania i uruchomienia programu. W katalogu „**Source**", plik **loop_intr.c** zawiera kod źródłowy programy napisany w języku C.

Kompilacja programu polega na wybraniu opcji: **Project – Build.** Jeżeli kod programu nie zawiera błędów wówczas po kompilacji zostanie wydany komunikat:

- *Build Complete, 0 Errors, 0 Warnings, 0 Remarks,*

oraz utworzony plik wykonywalny dla procesora o nazwie **loop_intr.out**

Plik **loop_intr.out** należy **przesłać do DSK** przez wybranie opcji: **File – Load program..** (wybierz plik: **loop_intr.out**), następuje ładowanie



Ostatnim krokiem jest **uruchomienie** programu, po wybraniu opcji **Debug – Run** (zatrzymanie: Debug – Halt). Aby sprawdzić poprawność działania programu ustaw na generatorze sygnał wejściowy dla DSK jako przebieg sinusoidalny o częstotliwości 1 kHz i amplitudzie 1 V. Obserwuj na oscyloskopie sygnał wejściowy i wyjściowy DSK, oba przebiegi powinny być jednakowe, ew. przesunięte w fazie.

(30)

ĆWICZENIE 1

Poniżej przedstawiono kod programu, który w tym ćwiczeniu wykonuje procesor sygnałowy

```
//Loop_intr.c
//W programie próbki wejściowe są przekazywane na wyjście układu DSK,
//
interrupt void c_int11()      //program obsługi przerwań
{
int sample_data;
sample_data = input_sample(); //dane wejściowe
output_sample(sample_data);   //dane wyjściowe
return;
}
void main()
{
comm_intr(); //inicjacja DSK
while(1);    //pętla nieskończona
}
```

W ćwiczeniu po skompilowaniu i uruchomieniu programu:

a) ***Sprawdzić jak zachowuje się układ przy zwiększaniu częstotliwości sygnału ponad częstotliwość 4 kHz. (częstotliwość próbkowania wynosi 8kHz) Na wejściu DSK znajduje się filtr antyaliasingowy: wyznacz jego charakterystykę? Czy można w tym układzie zaobserwować zjawisko aliasingu?***

b) ***Wyznaczyć zależność przesunięcia fazowego między sygnałem wejściowym i wyjściowym w zależności od częstotliwości, w zakresie 1-3kHz.***

c) ***Zwiększaj amplitudę sygnału wejściowego od 1,5V do 2,5V (nie więcej!!!!). Obserwuj sygnał wyjściowy DSK. Jak można wytłumaczyć to zjawisko?***

Po wykonaniu badań zatrzymaj program i **zamknij projekt1** opcje: **Project-Close-**

2.1. PROJEKT GENERATOR - TABLICA

W projekcie tym procesor sygnałowy jest wykorzystywany jako generator przebiegu okresowego. Jako przykład przedstawiono program generatora sinusa o częstotliwości 1kHz. W tablicy sine_table[] zapisano 8 wartości funkcji sinus w skali 1000 obliczone dla 1 okresu.

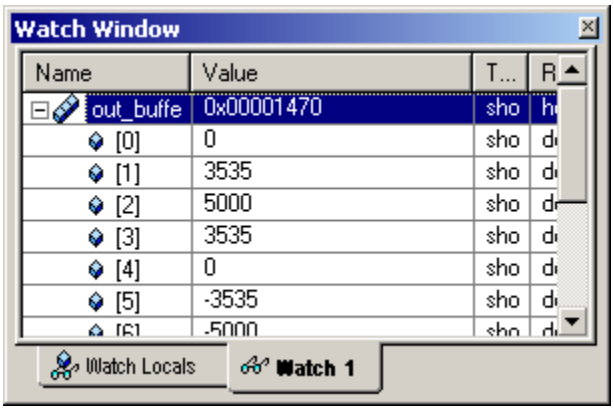
Co okres próbkowania równy 1/8000 = 0.125s, każda próbka z tablicy jest wysyłana na wyjście DSK, output_sample(). Czyli okres generowanego przebiegu wynosi 8*0.125s=1ms.

Zakres amplitudy sygnału wyjściowego jest ograniczony liczbą bitów przetwornika C/A i zawiera się w granicach od –32768 do +32767

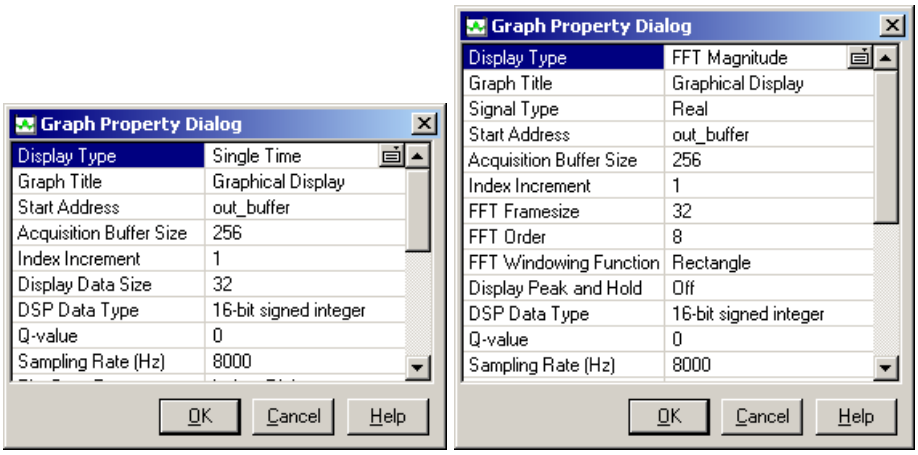
Próbki generowanego sygnału są dodatkowo pamiętane w buforze o rozmiarze 256, dzięki temu wartości te można monitorować w postaci tekstowej lub graficznej za pomocą CCS.

Wczytaj do CCS projekt **sine8_buf.pjt** z katalogu c:\ti\myprojects\projekt2\ oraz skompiluj i uruchom program **sine8_buf,** na oscyloskopie obserwuj przebieg wyjściowy DSK.

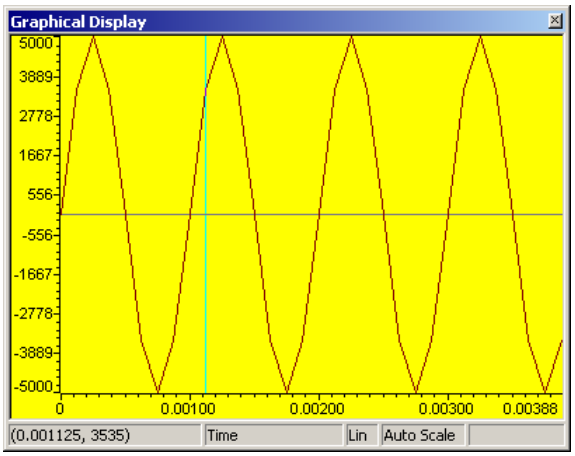
Aby monitorować wartości zmiennych programu w postaci tekstowej należy w CCS otworzyć okno „watch", opcja: **View – Watch window -** , a następnie w tym oknie w kolumnie „**Name**" wpisać nazwę zmiennej np. **out_buffer.** Jeżeli program jest uruchomiony w to pojawią się wartości próbek

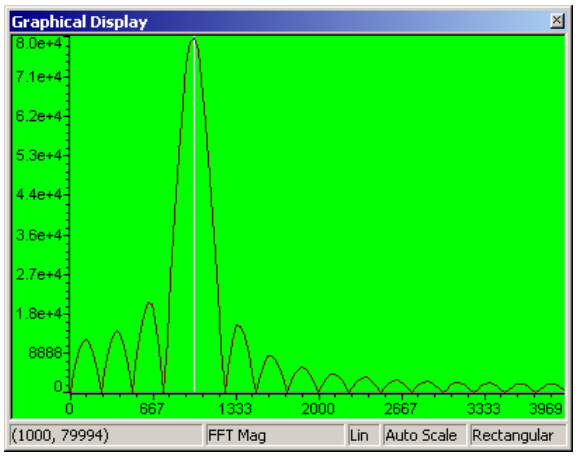


Aby przedstawić wartości próbek w postaci graficznej należy wybrać opcje: **View – Graph –Time/Frequency,** a następnie wybrać odpowiednie opcje okna graficznego jak poniżej, w zależności czy monitorujemy wartości próbek sygnału , czy jego widmo amplitudowe



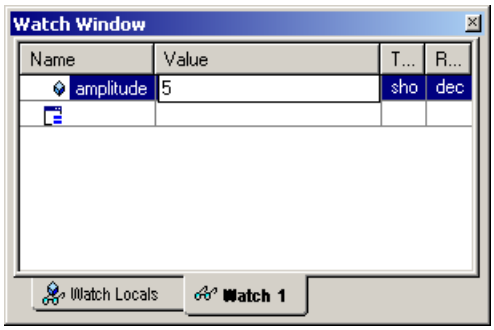
Wykresy poniżej wykonane w CCS pozwalają monitorować w czasie rzeczywistym zmienne programu wykonywanego przez procesor sygnałowy:



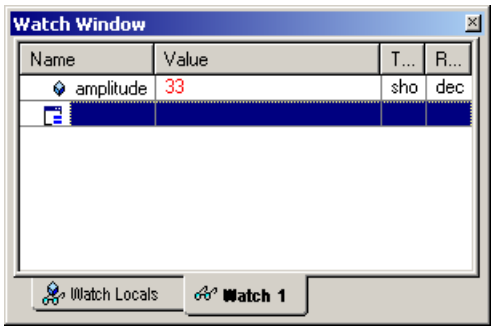


Środowisko CCS pozwala także na zmianę wartości parametrów zmiennych, zmianę taką można wykonać w oknie „**watch**”. Wprowadź w kolumnie „**Name**” nazwę zmiennej „**amplitude**”. Ponieważ w programie ta zmienna ma wartość 5, w kolumnie „**Value**”, taka wartość zostanie pokazana. Można ją zmienić podając inną wartość.

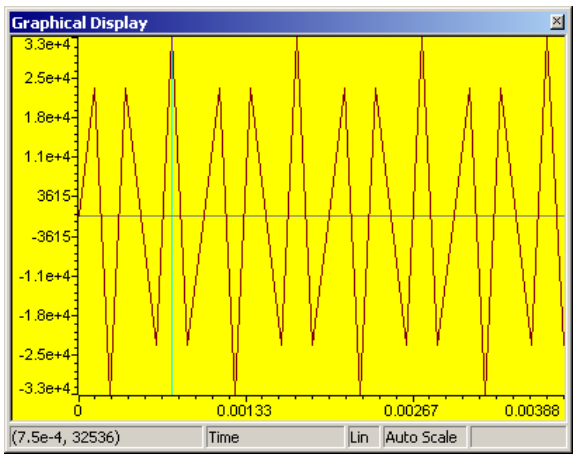
Przed

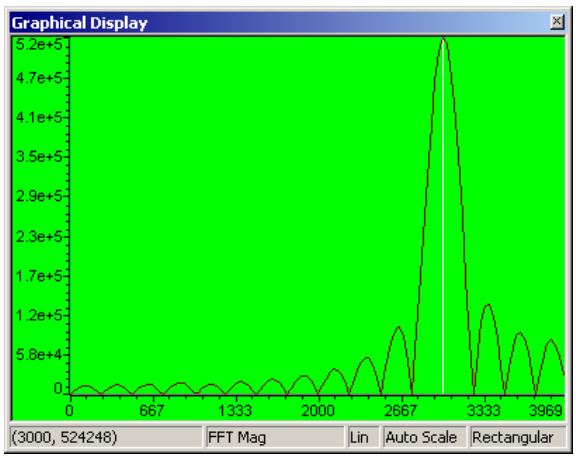


Po



Rysunki poniżej przedstawiają przypadek, gdy w wyniku dokonanej zmiany, wartość amplitudy sygnału przekroczyła dopuszczalną wartość dla liczby 16 bitowej, (33*1000>32768), sygnał wyjściowy jest zniekształcony. Porównaj z poprzednimi wykresami.





(60)

ĆWICZENIE 2.1

Poniżej przedstawiono kod programu:

```
//sine8_buf Generacja sinusa.
short loop = 0;
short sine_table[8] = {0,707,1000,707,0,-707,-1000,-707}; //wartości sinusa
w okresie
short out_buffer[256];              //bufor wyjściowy
const short BUFFERLENGTH = 256;     //rozmiar bufora wyjściowego
short i = 0;                        //indeks  bufora wyjściowego
short amplitude = 5;                //amplituda
interrupt void c_int11()            //program obsługi przerwań
{
  output_sample(amplitude*sine_table[loop]);  //próbka sinusoidy na wyjście
  out_buffer[i] = amplitude*sine_table[loop]; //bufor wyjściowy
  i++;                                        //zwiekszenie indeksu
  if (i == BUFFERLENGTH) i = 0;     //zerowanie indeksu gdy koniec
  if (loop < 7) ++loop;             //indeks tablicy sinusa
  else loop = 0;                    //jesli koniec tablicy sunusa,
  return;
}
void main()
{
  comm_intr();                    //inicjacja DSK
  while(1);                       //pętla nieskończona
}
```

W ćwiczeniu po skompilowaniu i uruchomieniu programu:

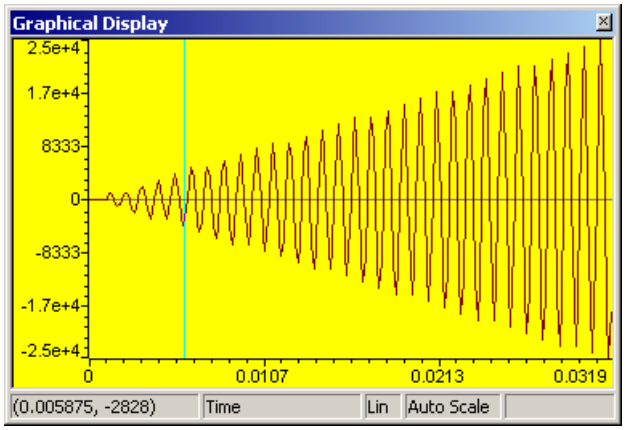
Ćw. 2.1

Gr.	A	f
1	8000	500
2	11000	500
3	14000	500
4	17000	800
5	20000	800
6	23000	800

a) ***Zmodyfikuj program tak aby procesor sygnałowy generował przebieg sinusoidalny o amplitudzie A oraz częstotliwości f.***

b) ***Wykonaj w CCS wykres zadanego (generowanego) przebiegu sygnału oraz jego widmo amplitudowe.***

c) ***Zmodyfikuj program w ten sposób aby amplituda przebiegu generowanego zmieniała się w czasie, np. tak jak na rysunku niżej. Pamiętaj o dopuszczalnym zakresie zmian amplitudy.***



Po wykonaniu badań zatrzymaj program i **zamknij projekt2** opcje: **Project-Close-**

2.2. PROJEKT GENERATOR - FUNKCJA

W projekcie programuje się generator przebiegu okresowego, jednak w odróżnieniu od poprzedniego projektu, tu wartości próbek są obliczane przy wykorzystaniu biblioteki funkcji matematycznych **math.h**. Jako przykład przedstawiono program generatora sinusa.

Wczytaj do CCS projekt **sinegen_table.pjt** z katalogu c:\ti\myprojects\projekt3\ oraz skompiluj i uruchom program **sinegen_table,** na oscyloskopie obserwuj przebieg wyjściowy DSK.

(90)

ĆWICZENIE 2.2

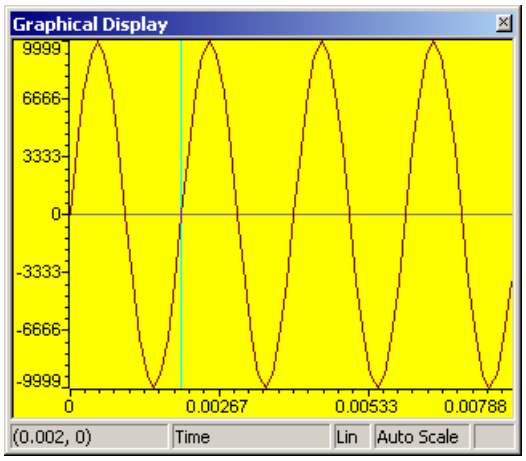
Obliczony i generowany przebieg sinusoidalny ma zadaną amplitudę równą 10000. Częstotliwość przebiegu wynosi 4*8000/64 =500 Hz (4 harmoniczna).

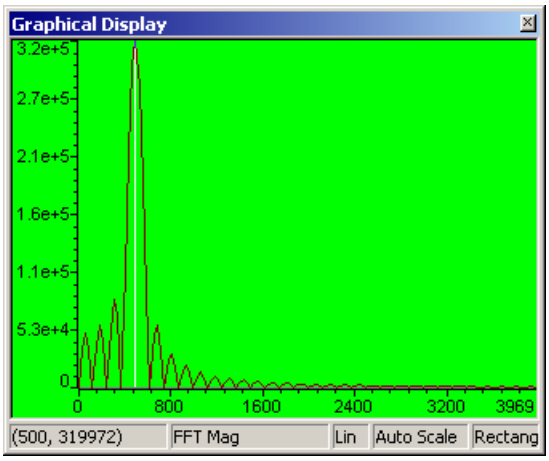
```
//Sinegen_table.c Generacja przebiegu w pętli nieskończonej
#include <math.h>
#define N 64                          //wymiar tablicy

short sine_table[N];                  //deklaracja  tablicy
short A = 10000;                      //wartość amplitudy

void main()
{float pi=3.1415;
  int i;
for (i=0;i<N;i++)                     // obliczenia wartości funkcji
                                      // generowanej
sine_table[i]=A*sin(2*pi*i/N);

  comm_poll();                        //inicjacja DSK
  i=0;
  while(1)                            //pętla nieskończona
  {
   output_sample(sine_table[i]);      //próbka wyjściowa
   if (i < N-1) i++;
   else i = 0;                        //zerowanie indeksu
  }
}
```





W ćwiczeniu po skompilowaniu i uruchomieniu programu:

Ćw. 2.2

Gr.	f1	f2	f3
1	125	250	500
2	250	375	625
3	500	625	750
4	125	375	750
5	250	625	875
6	500	750	1000

a) ***Zmodyfikuj program tak, aby procesor sygnałowy generował przebieg zawierający 3 składowe sinusoidalne o częstotliwościach f1,f2,f2 i dowolnych amplitudach.***

b) ***Wykonaj w CCS wykres zadanego (generowanego) przebiegu sygnału oraz wykres jego widma amplitudowego.***

c) ***Wygeneruj sygnał, którego wartości są obliczane z innych funkcji, niż sinusoidalne.***

3. PROJEKT- OPERACJA NA PRÓBKACH

W projekcie tym program przykładowy **echo.c** demonstruje sposób modyfikowania w czasie rzeczywistym wartości próbek wejściowych. Każda próbka wejściowa jest mnożona przez wartość kolejną wartość z tablicy sinusa i wynik podawany na wyjście. Następuje modulacja sygnału wejściowego przebiegiem sinusoidalnym. Częstotliwość przebiegu sinusoidalnego zależy od liczby wartości w tablicy. Dla czterech wartości wynosi ona 8000/4=2000 Hz

Wczytaj do CCS projekt **echo.pjt** z katalogu **C:\ti\myprojects\projekt4** oraz skompiluj i uruchom program **echo.** Na wejście DSK należy podaj sygnał o częstotliwości 50 Hz i amplitudzie 1 V, na oscyloskopie obserwuj zmodulowany przebieg wyjściowy.

(120)

ĆWICZENIE 3

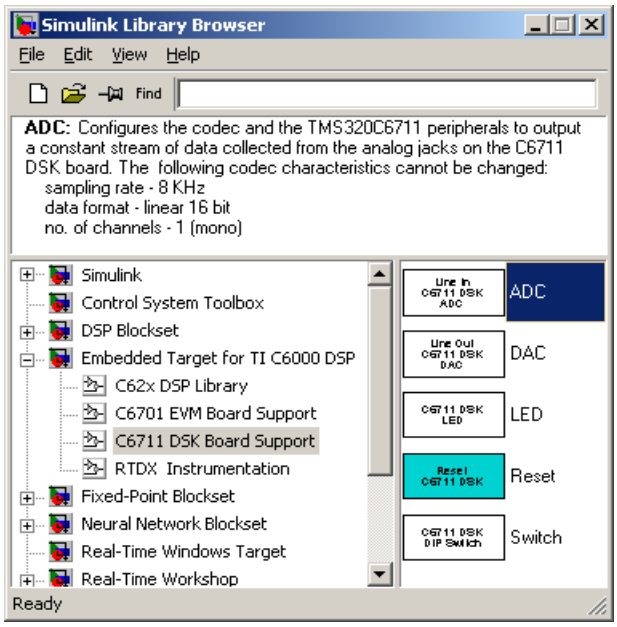
```
//Echo.c  Modulacja amplitudy
short input, output;
short loop = 0;
float sine_table[4] = {0,1,0,-1};     //wartości sinusa 2kHz
interrupt void c_int11()              //program obsługi przerwań
{
  input = input_sample();             // próbka wejściowa
  output=input*sine_table[loop];      // próbka wyjściowa = wejściowa * sinus
  output_sample(output);              // próbka wyjściowa
  if (loop < 4) ++loop;               // indeks tablicy sinusa 2kHz
  else loop = 0;                      // zeruj indeks tablicy
  return;
}
void main()
{
  comm_intr();                       //inicjacja DSK
  while(1);                          //pętla nieskończona
}
```

a) ***Zmieniaj rozmiar tablicy „sine_table, jaką częstotliwość na sygnał modulujący (sinusoidalny?) obserwuj efekt.***

b) ***Zaproponuj inny, użyteczny algorytm operacji na próbkach, sprawdź poprawność algorytmu doświadczalnie.***

4. PROJEKT SIMULINK – TMS

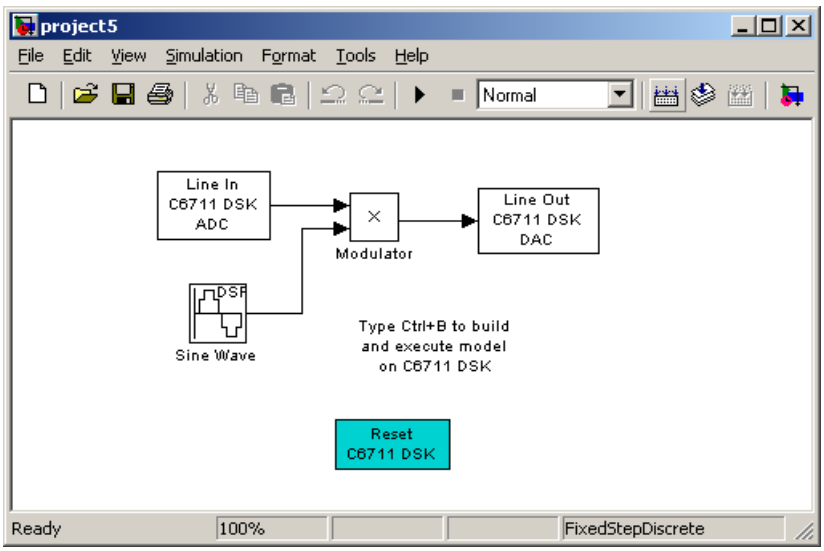
Alternatywnym sposobem programowania procesora TMS320C6711 jest SIMULINK wraz z dołączoną biblioteką. „**Embedded Target for TI C6000 DSP**”. Biblioteka zawiera między innymi modele wejścia i wyjścia układu DSK, a także szereg specjalnych funkcji cyfrowego przetwarzania sygnałów. Blokowa struktura projektu pozwala na wygodne i szybkie uruchamianie programów w układzie DSK.



Uruchom Matlaba, przejdź do katalogu **c:\ti\myprojects\projekt5** i uruchom SIMULINK’a:

```
>> cd  c:\ti\myprojects\projekt5\
>> pwd
>> simulink
```

Otwórz w SIMULINK’u plik **projekt5.mdl**



Wykonaj kompilację i uruchom program w DSK, wybierając na klawiaturze **Ctrl+B.**

Następuje konwersja modelu SIMULINK'a na język „C", tworzenie projektu dla CCS, uruchomienie CCS oraz wczytanie projektu, kompilacja projektu, przesłanie programu wykonywalnego do DSK i uruchomienie programu – czyli pełny automat !!!.

W **c:\ti\myprojects\projekt5** zostanie utworzony podkatalog ..\ projekt5_c6000_rtw\ zawierający kompletny projekt dla CCS.

(150)

ĆWICZENIE 4

a) Otwórz model projekt5.mdl w programie SIMULINK, odczytaj parametry wszystkich bloków w tym modelu. Uruchom model w DSK i obserwuj sygnały wejściowy i wyjściowy na oscyloskopie.

b) Zmień częstotliwość przebiegu sinusoidalnego poprzez zmianę wymiaru tablicy zawierającej wartości sinusa, obserwuj zmiany na oscyloskopie

c) Zaproponuj inny (własny) model cyfrowego przetwarzania sygnałów w SIMULINK'u, uruchom go w DSK i sprawdź poprawność działania na oscyloskopie.

(180)

UWAGA !

Ponieważ układ DSK mimo swojego dydaktycznego przeznaczenia lubi bez wyraźnego powodu czasami się zaciąć, jeżeli nie pomaga już nic innego, należy przeprowadzić procedurę zimnego startu płytki:

- ***Zirytuj się !!!***
- ***Zamknij projekt w CCS***
- ***Zamknij CCS***
- ***Wyłącz zasilanie DSK, poczekaj 1 sek.***
- ***Załącz zasilanie DSK***
- ***Uruchom CCS***
- ***Wczytaj projekt***
- ***Pracuj***

LITERATURA:

Rulph Chassaing "*Dsp Applications Using C & Tms320c6X DSK*" JOHN WILEY & SONS, INC.2002